This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims

1. (Currently Amended) A computer system, comprising:

a runtime operating environment that comprises a computer processor and an operating system adapted to execute:

a first object code portion containing instructions for eliciting certain actions from a Java virtual machine (JVM) associated with any of ~~the~~ a plurality of operating environments;

a second object code portion containing instructions for eliciting certain actions from the operating system associated with the runtime operating environment; and

first and second source code portions, wherein the first object code portion is derived from the first source code portion, the second object code portion is derived from the second source code portion, and wherein the second source code portion is adapted to derive a third object code portion containing instructions for eliciting certain actions from a second operating environment, different from the runtime operating environment.

2. (Original) The computer system as recited in claim 1, wherein execution of the instructions contained within the first and second object code portions may be initiated by a Java application program acting within the runtime operating environment.

3. (Original) The computer system as recited in claim 1, wherein the first source code portion comprises definitions of Java abstract windowing toolkit (AWT) Component Peer classes written in the Java programming language.

4. (Original) The computer system as recited in claim 1, wherein the second source code portion comprises definitions of native methods of Java AWT Component Peer classes written in the C/C++ programming language.

5. (Original) The computer system as recited in claim 1, wherein the second object code portion comprises a host platform interface (HPI) between the AWT Component Peer classes and the runtime operating environment.

A1

6. (Currently Amended) A method for ~~interfacing an application program to a runtime operating environment comprising a computer processor and an operating system, the method~~ creating an operating system independent programming interface that provides consistent presentation results across a plurality of operating systems, the method comprising:

~~deriving from a first source code portion a first object code portion containing instructions executable by the computer processor within the runtime operating environment for eliciting certain actions from a Java virtual machine (JVM) associated with the runtime operating environment;~~

~~deriving from a second source code portion a second object code portion containing instructions executable by the computer processor within the runtime operating environment for eliciting certain actions from the operating system within the runtime operating environment; and~~

~~deriving from the second source code portion a third object code portion containing instructions executable by a second computer processor within a second operating environment for eliciting certain actions from said second operating environment, wherein the second operating environment is different from the runtime operating environment~~

receiving first source code describing an operating system independent function, wherein said first source code includes an interface reference to an operating system dependent function;

transforming said first source code into first machine executable code, said first machine executable code operable on any of a plurality of operating systems having an operating system independent virtual execution function;

receiving second source code describing an operating system dependent function responsive to said interface reference; and

transforming said second source code into a plurality of machine executable code modules, each of said machine executable modules operable on only one of said plurality of operating systems.

7. (Currently Amended) The method as recited in claim 6, further comprising an active Java application program acting within ~~the~~ a runtime operating environment to initiate execution of the instructions contained within ~~the~~ first and second object code portions.

8. (Currently Amended) The method as recited in claim ~~6~~7, further comprising defining Java AWT Component Peer classes in the first source code ~~portion~~.

9. (Currently Amended) The method as recited in claim 6, wherein the first source code ~~portion~~ is written in the Java programming language.

10. (Currently Amended) The method as recited in claim 6, further comprising defining native methods of Java AWT Component Peer classes in the second source code ~~portion~~.

11. (Currently Amended) The method as recited in claim 6, wherein the second source code ~~portion~~ is written in the C/C++ programming language.

12. (Currently Amended) The method as recited in claim ~~6~~8, further comprising defining a host platform interface (HPI) between the AWT Component Peer classes and the runtime operating environment in the second object code portion.

13. (Original) A computer-readable storage device, comprising:

a runtime operating environment comprising computer processor and an operating system;

a first object code portion, containing instructions executable by the computer processor within the runtime operating environment for eliciting certain actions from a Java virtual machine (JVM) associated with the runtime operating environment;

a second object code portion, wherein the object code contains instructions executable by the computer processor within the runtime operating environment for eliciting certain actions from the operating system within the runtime operating environment; and

first and second source code portions, wherein the first object code portion is derived from the first source code portion, the second object code portion is derived from the second source code portion, and wherein the second source code portion is adapted to derive a third object code portion containing instructions executable by a second computer processor within a second operating environment for eliciting certain actions from said second operating environment, wherein the second operating environment is different from the runtime operating environment.

14. (Original) The computer-readable storage device as recited in claim 13, wherein the first source code portion comprises definitions of at least some of the Java AWT Component Peer classes.

15. (Original) The computer-readable storage device as recited in claim 13, wherein the second object code portion comprises a host platform interface (HPI) between the AWT Component Peer classes and the runtime operating environment.